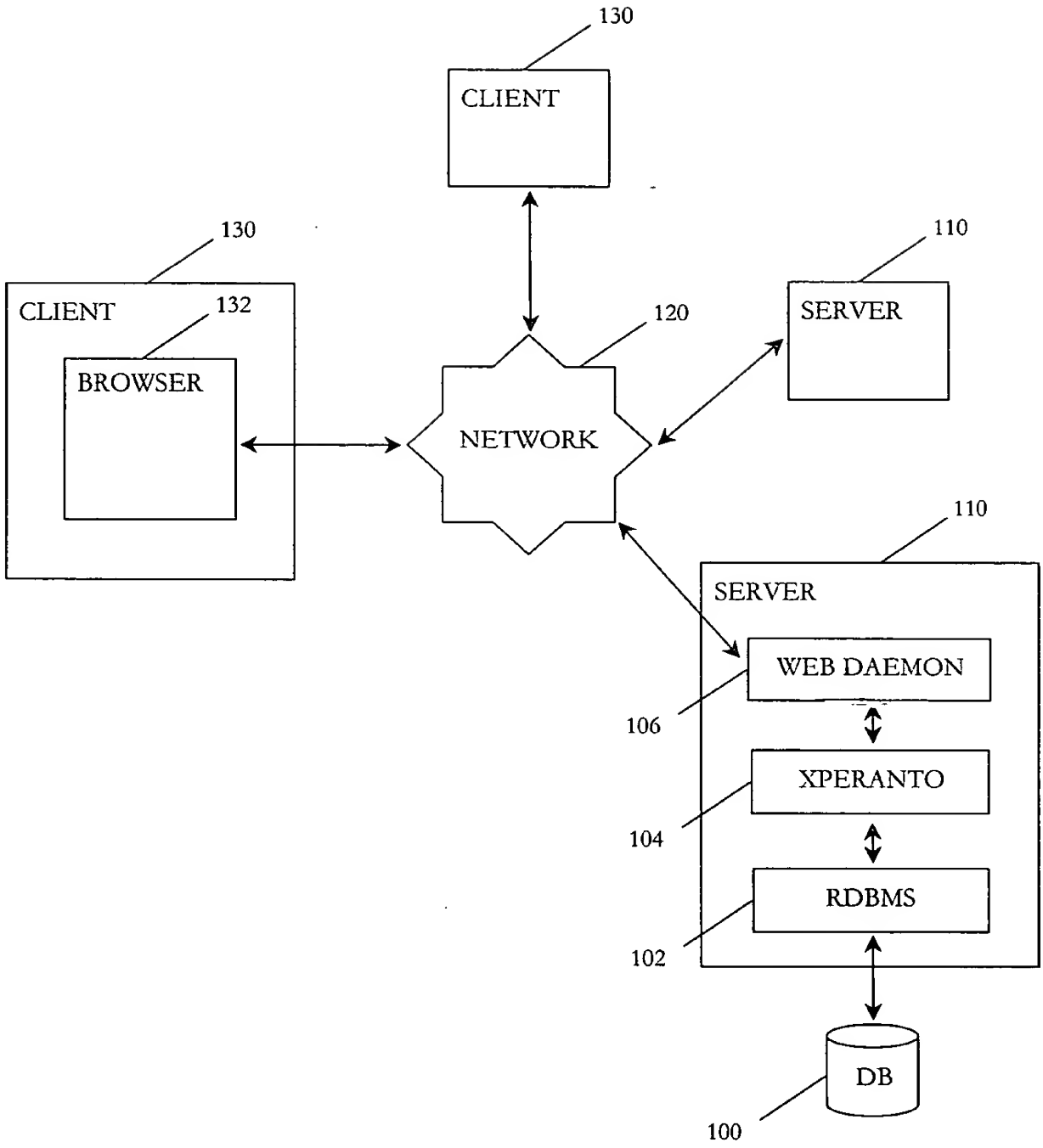
130
CLIENT
130
CLIENT
132
BROWSER
120
NETWORK
110
SERVER
110
SERVER
WEB DAEMON
106
XPERANTO
104
RDBMS
102
DB
100

FIG. 1

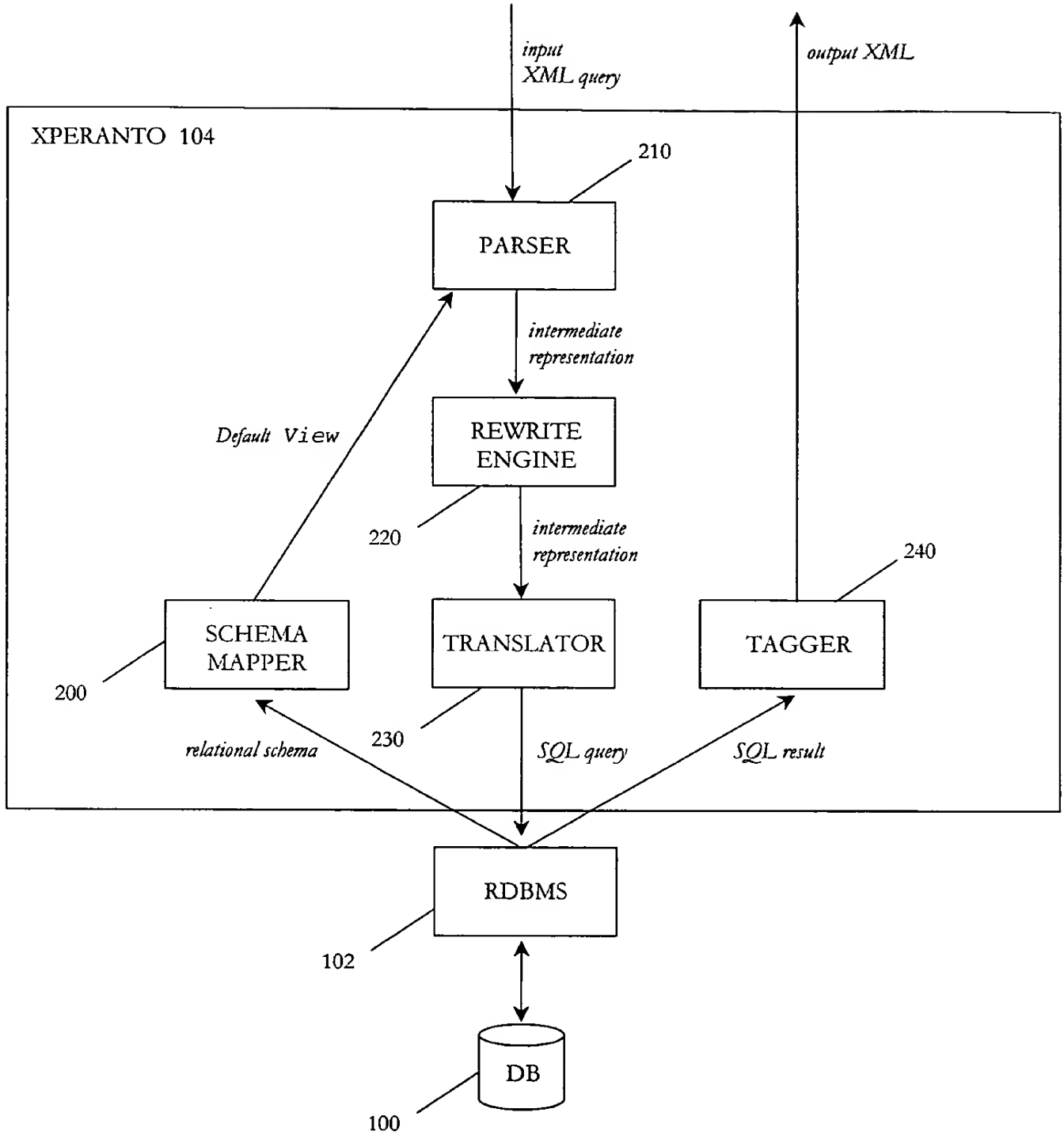
input
XML query
output XML
XPERANTO 104
210
PARSER
intermediate
representation
Default View
REWRITE
ENGINE
220
intermediate
representation
240
SCHEMA
MAPPER
TRANSLATOR
TAGGER
200
230
relational schema
SQL query
SQL result
RDBMS
102
DB
100

FIG. 2

DEPT TABLE 300

DNO	NAME	SIZE
1	Finance	100
2	Marketing	200
3	Sales	50
...		

EMP TABLE 310

DNO	ENO	NAME	AGE
1	1	J. Skolem	50
2	2	L. Hubbard	33
...			

Default XML View of the DEPT Table 320

```
<DEPT>
  <ROW>
    <DNO> 1 </DNO> <NAME> Finance </NAME> <SIZE> 100 </SIZE>
  </ROW>
  <ROW>
    <DNO> 2 </DNO> <NAME> Marketing </NAME> <SIZE> 200 </SIZE>
  </ROW>
  <ROW>
    <DNO> 3 </DNO> <NAME> Sales </NAME> <SIZE> 50 </SIZE>
  </ROW>
  ...
</DEPT>
```

Default XML View of the EMP Table 330

```
<EMP>
  <ROW>
    <DNO> 1 </DNO> <ENO> 1 </ENO> <NAME> J. Skolem </NAME> <AGE> 50 </AGE>
  </ROW>
  <ROW>
    <DNO> 2 </DNO> <ENO> 2 </ENO> <NAME> L. Hubbard </NAME> <AGE> 33 </AGE>
  </ROW>
  ...
</EMP>
```

FIG. 3

Query Over the Default View

```
WHERE
  <DEPT>
    <ROW>
      <DNO> $DNO </DNO> <NAME> $DNAME </NAME>
      <SIZE> $DSIZE </SIZE>
    </ROW>
  </DEPT> IN DefaultView,
  $DSIZE > 75
CONSTRUCT
  <BIGDEPT>
    <NAME> $DNAME </NAME>
    <SIZE> $DSIZE </SIZE>
    {WHERE
      <EMP>
        <ROW> <DNO> $DNO </DNO> <NAME> $ENAME </NAME> </ROW>
      </EMP> IN DefaultView
     CONSTRUCT
      <EMP> <NAME> $ENAME </NAME> </EMP>}
  </BIGDEPT>
```

XML Produced by the Query

```
<BIGDEPT>
  <NAME> Finance </NAME>
  <SIZE> 100 </SIZE >
  <EMP> <NAME> J. Skolem </NAME> </EMP>
  ...
</BIGDEPT>
<BIGDEPT>
  <NAME> Marketing </NAME>
  <SIZE> 200 </SIZE >
  <EMP> <NAME> L. Hubbard </NAME> </EMP>
  ...
</BIGDEPT>
...
```

FIG. 4

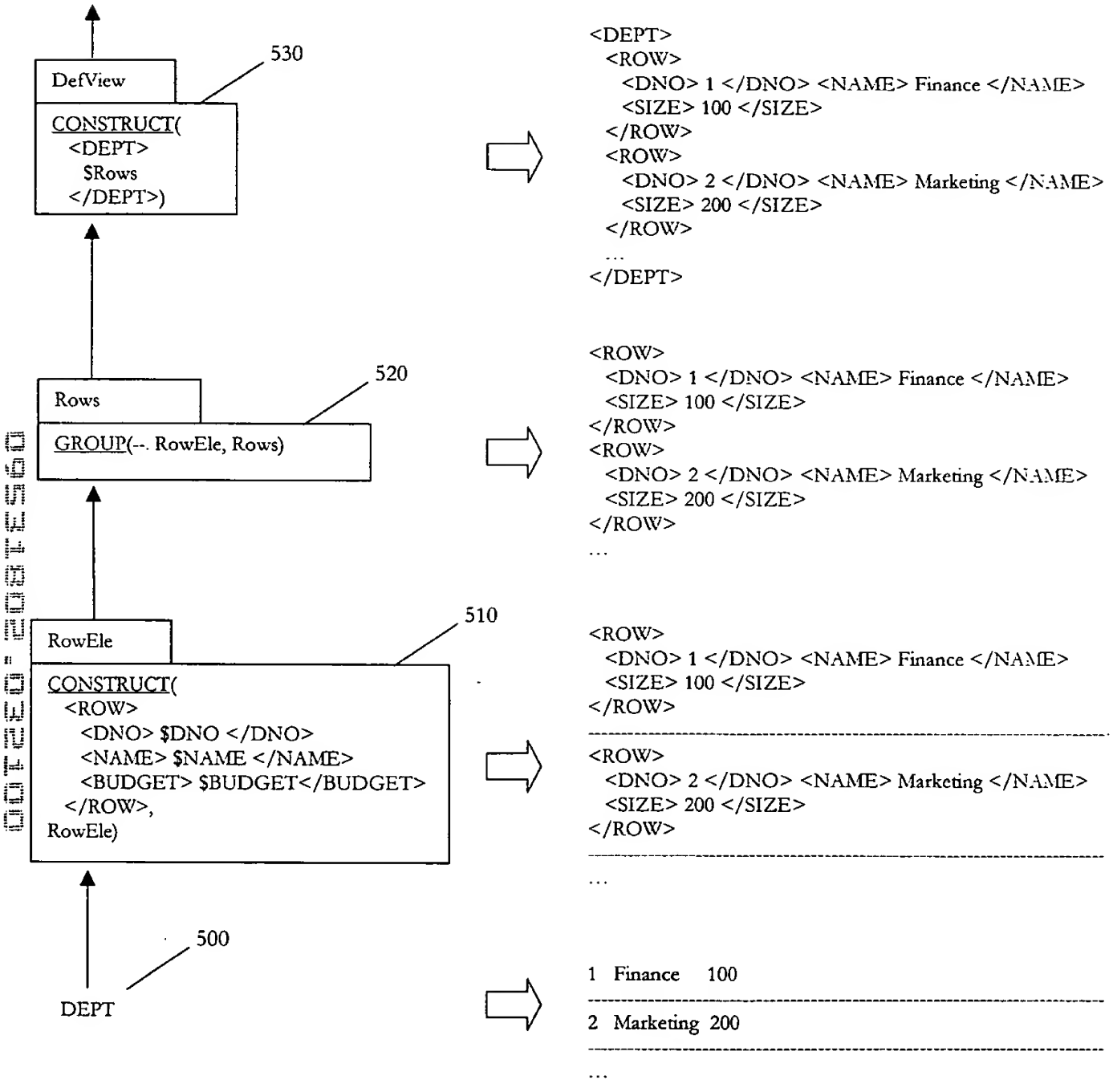
Intermediate Representation for the Default View of the DEPT Table
Output of Each Operation
DefView
530
CONSTRUCT(
<DEPT>
$Rows
</DEPT>)
Rows
520
GROUP(--. RowEle, Rows)
RowEle
510
CONSTRUCT(
<ROW>
<DNO> $DNO </DNO>
<NAME> $NAME </NAME>
<BUDGET> $BUDGET</BUDGET>
</ROW>,
RowEle)
DEPT
500
<DEPT>
<ROW>
<DNO> 1 </DNO> <NAME> Finance </NAME>
<SIZE> 100 </SIZE>
</ROW>
<ROW>
<DNO> 2 </DNO> <NAME> Marketing </NAME>
<SIZE> 200 </SIZE>
</ROW>
...
</DEPT>
<ROW>
<DNO> 1 </DNO> <NAME> Finance </NAME>
<SIZE> 100 </SIZE>
</ROW>
<ROW>
<DNO> 2 </DNO> <NAME> Marketing </NAME>
<SIZE> 200 </SIZE>
</ROW>
...
<ROW>
<DNO> 1 </DNO> <NAME> Finance </NAME>
<SIZE> 100 </SIZE>
</ROW>
<ROW>
<DNO> 2 </DNO> <NAME> Marketing </NAME>
<SIZE> 200 </SIZE>
</ROW>
...
1 Finance 100
2 Marketing 200
...

FIG. 5

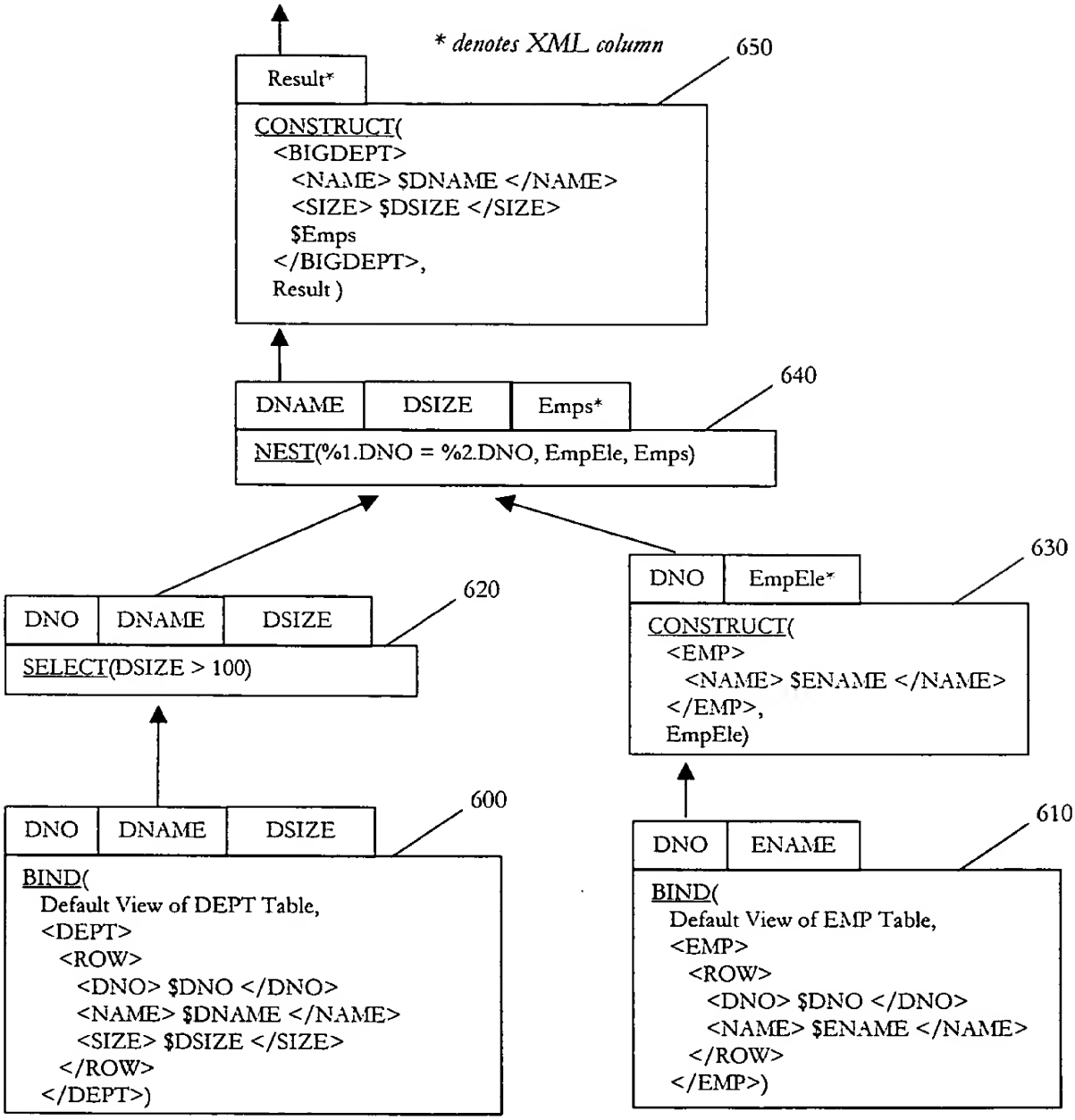
* denotes XML column
650
Result*
CONSTRUCT(
<BIGDEPT>
<NAME> $DNAME </NAME>
<SIZE> $DSIZE </SIZE>
$Emps
</BIGDEPT>,
Result)
640
DNAME
DSIZE
Emps*
NEST(%1.DNO = %2.DNO, EmpEle, Emps)
630
DNO
EmpEle*
CONSTRUCT(
<EMP>
<NAME> $ENAME </NAME>
</EMP>,
EmpEle)
620
DNO
DNAME
DSIZE
SELECT(DSIZE > 100)
600
DNO
DNAME
DSIZE
BIND(
Default View of DEPT Table,
<DEPT>
<ROW>
<DNO> $DNO </DNO>
<NAME> $DNAME </NAME>
<SIZE> $DSIZE </SIZE>
</ROW>
</DEPT>)
610
DNO
ENAME
BIND(
Default View of EMP Table,
<EMP>
<ROW>
<DNO> $DNO </DNO>
<NAME> $ENAME </NAME>
</ROW>
</EMP>)

FIG. 6

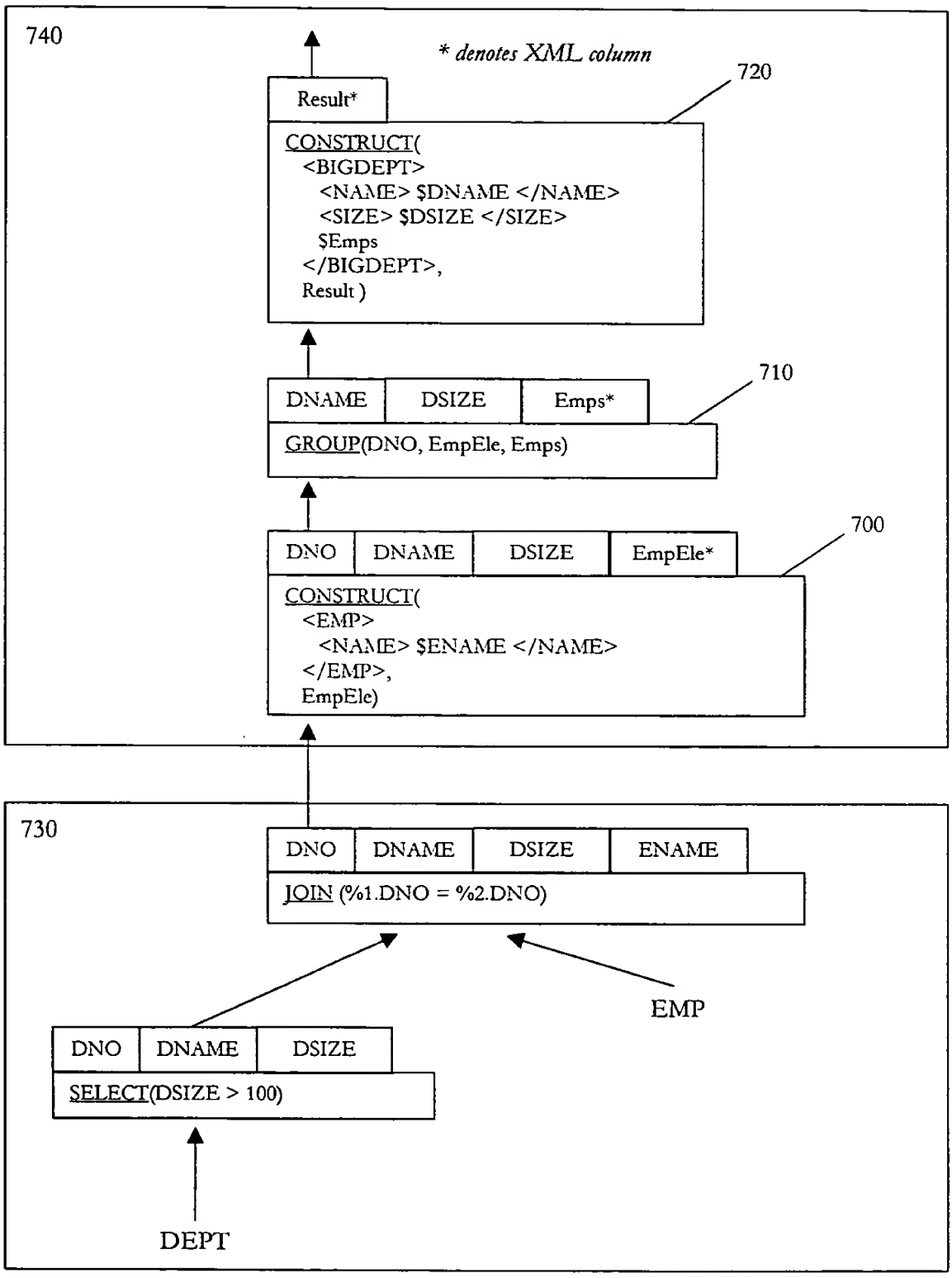
740
* denotes XML column
720
Result*
CONSTRUCT(
<BIGDEPT>
<NAME> $DNAME </NAME>
<SIZE> $DSIZE </SIZE>
$Emps
</BIGDEPT>,
Result)
710
DNAME
DSIZE
Emps*
GROUP(DNO, EmpEle, Emps)
700
DNO
DNAME
DSIZE
EmpEle*
CONSTRUCT(
<EMP>
<NAME> $ENAME </NAME>
</EMP>,
EmpEle)
730
DNO
DNAME
DSIZE
ENAME
JOIN (%1.DNO = %2.DNO)
EMP
DNO
DNAME
DSIZE
SELECT(DSIZE > 100)
DEPT

FIG. 7

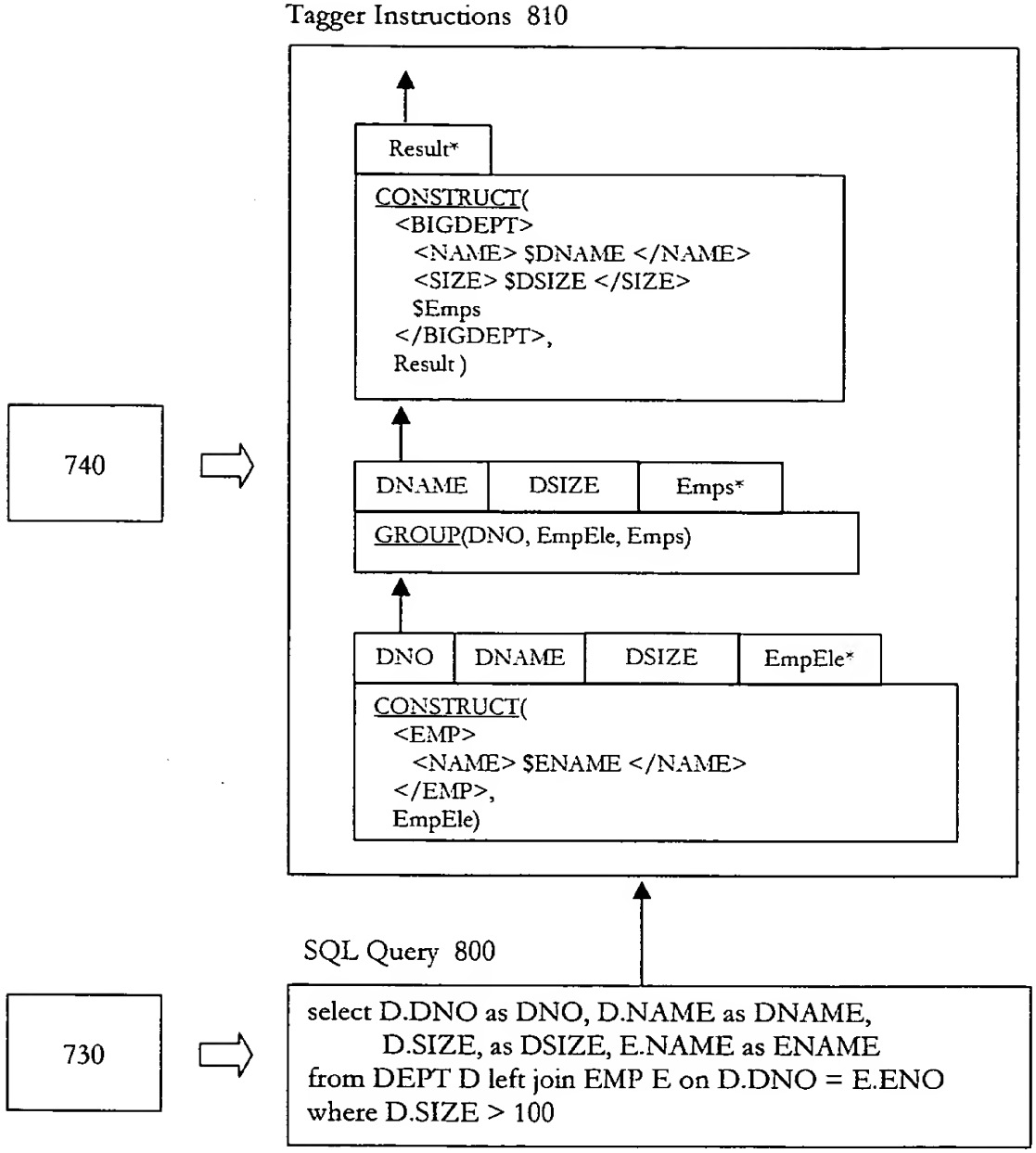
Tagger Instructions 810
Result*
CONSTRUCT(
<BIGDEPT>
<NAME> $DNAME </NAME>
<SIZE> $DSIZE </SIZE>
$Emps
</BIGDEPT>,
Result)
740
DNAME
DSIZE
Emps*
GROUP(DNO, EmpEle, Emps)
DNO
DNAME
DSIZE
EmpEle*
CONSTRUCT(
<EMP>
<NAME> $ENAME </NAME>
</EMP>,
EmpEle)
SQL Query 800
730
select D.DNO as DNO, D.NAME as DNAME,
D.SIZE, as DSIZE, E.NAME as ENAME
from DEPT D left join EMP E on D.DNO = E.ENO
where D.SIZE > 100

FIG. 8

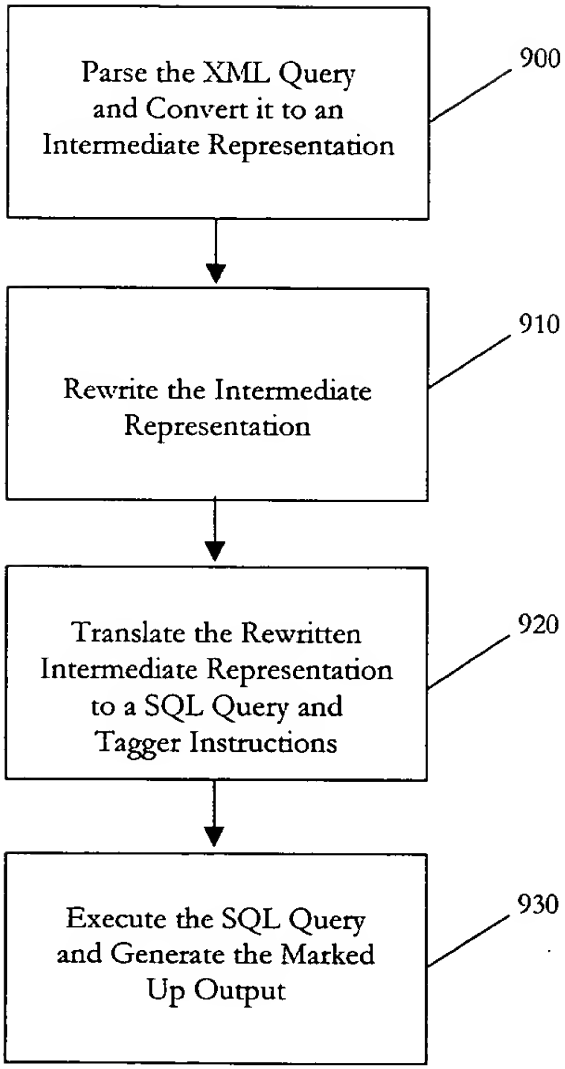
Parse the XML Query and Convert it to an Intermediate Representation
900
Rewrite the Intermediate Representation
910
Translate the Rewritten Intermediate Representation to a SQL Query and Tagger Instructions
920
Execute the SQL Query and Generate the Marked Up Output
930

FIG. 9